# Apply filters to SQL queries

Don Wickramanayake

## Project description

As a security professional at a large organization, part of our job is to investigate security issues to help keep the system secure. When we discover potential security issues that involve login attempts and employee machines our task is to examine the organization’s data in their employees and **log\_in\_attempts** tables. You’ll need to use SQL filters to retrieve records from different datasets and investigate the potential security issues. This project will go over some sample queries we can use to examine data in employee tables.

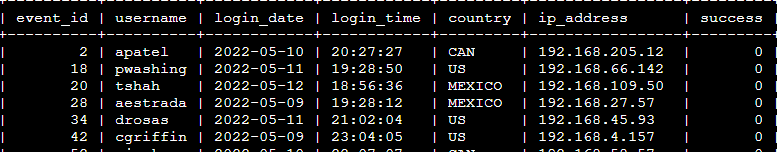
## Retrieve after hours failed login attempts

## When investigating failed login attempts that were made after business hours. You want to retrieve this information from the login activity. You’ll identify all unsuccessful attempts after 18:00.

SELECT \*

FROM log\_in\_attempts

WHERE login\_time > '18:00' AND success = FALSE;



## Retrieve login attempts on specific dates

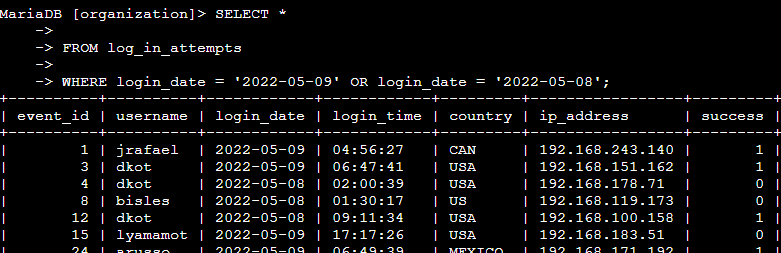
## When investigating a suspicious event that occurred on '2022-05-09'. You want to retrieve all login attempts that occurred on this day and the day before ('2022-05-08'). The **login\_date** column in the **log\_in\_attempts** table contains information on the dates when login attempts were made. Use the **OR** operator to retrieve the failed login attempts on the specified days.

SELECT \*

FROM log\_in\_attempts

WHERE login\_date = '2022-05-09' OR login\_date = '2022-05-08';

## 



## Retrieve login attempts outside of Mexico

When investigating logins that did not originate in Mexico, and you need to find this information. Note that the country field includes entries with 'MEX' and 'MEXICO'. You should use the **NOT** and **LIKE** operators and the matching pattern 'MEX%'.

LIKE

You can also filter based on a pattern. For example, you can identify entries that start or end with a certain character or characters. Filtering for a pattern requires incorporating two more elements into your WHERE clause:

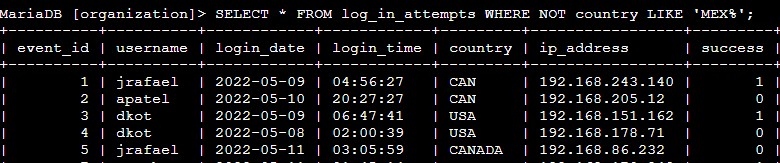
* a wildcard
* the LIKE operator

\*LIKE is used with WHERE to search for a pattern in a column.

SELECT \*

FROM log\_in\_attempts

WHERE NOT country LIKE 'MEX%';



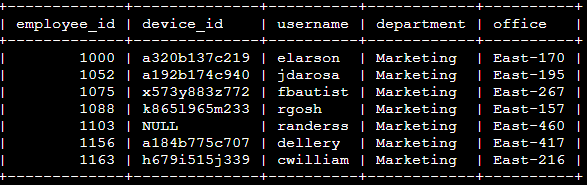
## Retrieve employees in Marketing

## Assume we are updating employee machines, and we need to obtain the information about employees in the 'Marketing' department who are located in all offices in the East building (such as 'East-170' or 'East-320').

SELECT \*

FROM employees

WHERE department = 'Marketing' AND office LIKE 'East%';



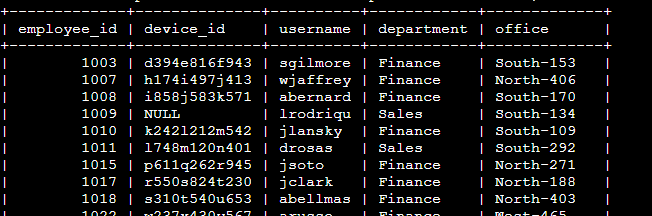
## Retrieve employees in Finance or Sales

## To perform a different update to the computers of all employees in the Finance or the Sales department, and you need to locate information on these employees. SQL query to retrieve records for employees in the 'Finance' or the 'Sales' department.

SELECT \*

FROM employees

WHERE department = 'Finance' OR department = 'Sales';



## Retrieve all employees not in IT

## This update was already made to employee computers in the Information Technology department. The team needs information about employees who are not in that department. You should use the **NOT** operator to identify these employees.

SELECT \*

FROM employees

WHERE NOT department = 'Information Technology';



## Summary

Using logical operators such as **AND, OR, NOT** we can filter out specific information that will help us to get data in security analysis. Also, we can use operators such as **>, <, =** for further filtering information in data tables. The **BETWEEN** operator, another inclusive operator, helps us return the data you need within a range.